

**South African  
Computer  
Journal  
Number 11  
May 1994**

**Suid-Afrikaanse  
Rekenaar-  
tydskrif  
Nommer 11  
Mei 1994**

**Computer Science  
and  
Information Systems**

**Rekenaarwetenskap  
en  
Inligtingstelsels**

**The South African  
Computer Journal**

*An official publication of the Computer Society  
of South Africa and the South African Institute of  
Computer Scientists*

**Die Suid-Afrikaanse  
Rekenaartydskrif**

*'n Amptelike publikasie van die Rekenaarvereniging  
van Suid-Afrika en die Suid-Afrikaanse Instituut  
vir Rekenaarwetenskaplikes*

---

**Editor**

Professor Derrick G Kourie  
Department of Computer Science  
University of Pretoria  
Hatfield 0083  
Email: dkourie@dos-lan.cs.up.ac.za

**Subeditor: Information Systems**

Prof John Shochot  
University of the Witwatersrand  
Private Bag 3  
WITS 2050  
Email: 035ebrs@witsvma.wits.ac.za

**Production Editor**

Dr Riël Smit  
Mosaic Software (Pty) Ltd  
P.O.Box 16650  
Vlaeberg 8018  
Email: gds@cs.uct.ac.za

---

**Editorial Board**

Professor Gerhard Barth  
Director: German AI Research Institute

Professor Pieter Kritzingen  
University of Cape Town

Professor Judy Bishop  
University of Pretoria

Professor Fred H Lochovsky  
University of Science and Technology, Kowloon

Professor Donald D Cowan  
University of Waterloo

Professor Stephen R Schach  
Vanderbilt University

Professor Jürg Gutknecht  
ETH, Zürich

Professor Basie von Solms  
Rand Afrikaanse Universiteit

---

**Subscriptions**

	Annual	Single copy
Southern Africa:	R45,00	R15,00
Elsewhere:	\$45,00	\$15,00

to be sent to:

*Computer Society of South Africa  
Box 1714 Halfway House 1685*



# **The FRD Special Programme on Collaborative Software Research and Development Draft Call for Proposals**

**Tony Krzesinski**

*Department of Computer Science, University of Stellenbosch, Stellenbosch, 7600  
aek1@cs.sun.ac.za*

## **Abstract**

*The Foundation for Research Development (FRD) is evaluating the need for a new Special Programme in Computer Science. An initial meeting was held in August 1993 at the FRD to discuss the new programme. A Steering Committee was appointed with Professor A E Krzesinski of Stellenbosch as Chairperson. Several regional meetings were held in the Western Cape, Eastern Cape and the PWV to discuss the parameters of the proposed programme. The result of these meetings was a Draft Call for Proposals which was discussed by the Steering Committee in January 1994 and which was further refined during February 1994.*

*The Steering Committee and the FRD have distributed the Draft Call to as wide an audience as possible. We hope that its publication in SACJ will facilitate the further spread of knowledge about the proposed programme.*

*The FRD wishes the programme to be collaborative, goal-directed, inter-institutional, inter-disciplinary, and definitive in the sense that it will set a direction in Computer Science in South Africa for the next five years. The goal of the programme is to link up research, development, industry and community organisations to develop expertise and manpower in order to establish a national capability for developing innovative world-class software.*

## **1 Summary**

The goal of the FRD Special Programme on Collaborative Software Research and Development (ColSoft) is to develop expertise and manpower in order to establish a national capability for developing innovative world-class software that will be self-sustaining at the conclusion of the ColSoft programme. The programme will use and develop key technologies, methodologies and expertise applicable to the software R&D process. The programme will promote regional and national collaboration and will establish, equip and develop collaborating groups of researchers and software developers at Universities, Technikons and participating Industry and Community organizations.

computing community to demonstrate their relevance and their ability to respond to the needs of the changed country and its people.

## **2 Scope**

The production of globally competitive products increasingly relies on the effective use of information technology: computers, networks, and software. Software is the key that unlocks the potential of computers and networks. Software enables people to use the hardware to do useful things. South Africa cannot achieve local development and global competitiveness unless we master the software skills.

component will develop theoretical and practical methods for the production of complex software systems.

All projects in the ColSoft programme will produce prototypes of innovative software systems that are usable and useful. The programme will develop human resources of a high level of scientific and technical excellence. The applications themselves will yield additional benefits including product development, tackling of community and business needs and potential for stimulating employment.

### 3 Focus

Successful research is generic in its applicability. Success in software research can be measured by the range of new and useful applications derived, by the increase in productivity of the users and by the power and proven applicability of the software tools produced.

#### Key Technologies

The ColSoft programme will focus on the following key fields of computer science that are relevant to the collaborative software research and development:

- databases
- distributed systems
- human-computer interaction
- network and system resources required for computer supported collaborative application development
- innovative tools to support quality software engineering
- support for the specification and design of large complex systems.

#### Key Applications

The ColSoft applications will address national needs such as communications, community development, education, environment, health, manufacturing, resource management and tourism.

#### Regional and National Collaboration

Producing innovative software will be a joint venture involving Universities and Technikons working with Industry and Community organizations. Collaboration among the programme participants will be encouraged, firstly because large teams of researchers and developers will be necessary to develop large and complex software systems, and secondly because the formation of consortia will allow teams to make best use of the different skills present among participating universities, technikons and industrial partners.

A substantial proportion of the ColSoft funds will therefore be allocated to consortia of researchers and developers. Funds will be explicitly allocated for fostering cooperation, for organizational support of viable consortia and for providing a suitable development environment for supporting remote cooperation. Each consortium will have a sound management structure and a clear and limited set of goals that are directed towards developing real products. A consortium may undertake several projects which must

be related to each other. The responsibilities of the team members will be clearly stated.

It is envisaged that the new generation of high speed networks will play a large role in the collaboration infrastructure where computer science itself is in the forefront of developing tools for remote collaboration.

#### Product Prototypes

The only convincing way to verify claims about useful computer applications is to build prototype systems and to evaluate their effectiveness. All projects in the ColSoft programme will produce working prototypes of innovative software systems that are usable and useful. The prototypes will reach beyond what is currently theoretically predictable and will be used to investigate how these systems function.

#### Communal Development Methodology

The ColSoft programme will provide a set of common national standards and a national software development methodology — such mutual support will be of benefit to all. National Internet forums will be used to enable co-operation between groups. The tools developed as part of the programme will be tested and used by programme application developers.

### 4 Objectives

The goal of the ColSoft programme is to develop expertise and manpower in order to establish a national capability for developing innovative world-class software that will be self-sustaining at the conclusion of the ColSoft programme.

#### Human Resource Development

Effective scientific exchanges and collaboration are only possible between peers. The ColSoft programme will develop human resources of a high level of scientific and technical excellence. Such local expertise of international quality is necessary if we are to draw upon and benefit from the rapid development of information technology. Thus although South Africa contributes only a small portion of the global research done in science and technology, we will be able to tap the enormous and world-wide research effort in computer technology through international cooperation.

#### Appropriate Technology

The ColSoft programme will address relative underdevelopment and enable participants to leap-frog several stages of development and avoid expensive blind alleys. Advanced information technology often provides solutions for a range of difficult problems: for example, the use of wireless and cellular technology provide cost effective telecommunications in rural and low infrastructure areas. Advanced information technology enables and supports other technological activities such as competitive manufacturing. The new high speed networks support effective computer mediated collaborative work. Communication

networks can provide essential support for community development.

#### **Innovation for Local Software Requirements**

The ColSoft programme will address local needs which might otherwise be neglected by international computer science research.

#### **Niches in the International Market**

The ColSoft programme should lead to an entry in international software markets via quality niche products. The success of South African software will open up a new export industry which makes extensive use of trained persons and requires a relatively small capital investment. Software development is particularly attractive because it can be practised on both a large ("factory" scale and on the scale of a small startup company.

## **5 Criteria**

#### **Application Criteria**

The ColSoft programme will fund research projects on the basis of the following criteria:

1. The research results must demonstrate a potential for addressing commercial, industrial or community needs through the development of prototypes.
2. The research component of the project must have an impact on relevant fields of computer science and result in improved software development expertise and infrastructure.
3. The project must demonstrate scientific merit and be of an innovative nature. The project must yield publications in recognised journals and conference proceedings, skilled graduates, and new developments in software applications.
4. The project must have a high chance of successful completion. The proposers must have an appropriate track record, or show clear potential.
5. The proposal must be technically sound and the human and equipment resources required to execute the project must be adequate.
6. The project will involve neglected sectors of the tertiary education system through, amongst others, the formation of consortia with established graduate schools.

#### **Commonality with Other FRD Programmes**

A project will not be disqualified because it addresses software development issues in areas supported by other FRD programmes. Demonstrable synergy (not duplication) with other research projects will contribute to the significance of a project.

#### **Hardware and Software Requirements**

The final software product must use equipment that will be affordable and generally available to the intended users when the development is complete. Equipment and sup-

port software should preferably be supplied by industrial partners or be funded under existing FRD schemes such as the multi-user equipment programme. The FRD recognises that ColSoft projects may need advanced specialised equipment and support for prototype development. The ColSoft programme will support purchases of equipment and development software if a need is clearly shown. Computer equipment such as currently standard personal computers and workstations are unlikely to be funded.

#### **Temporary Staff**

Funds will be available to appoint temporary and part-time technical programming staff needed to produce the prototype software systems which this programme demands. Technikon students as part of their practical training may be appointed as technical programmers to participating consortia.

#### **Exceptions**

There may be pressing needs in software research and development that are not covered by the above formulation of the ColSoft programme. Support will be given to such special projects only if the project is shown to be urgent and relevant and will foster software research and development and will be unlikely to be funded from other sources.

## **6 Evaluation**

The ColSoft programme will commence in January 1995 and will run for a period of five years. Participants will submit annual reports and present interim results at annual national meetings to which participants, industrial and community organisations are invited. The final evaluation will consist of written reports, together with presentations and demonstration of the system at a conference. Concrete utilization plans for the products and user evaluations of prototypes will form part of the requirements.

## **7 Funding**

It is anticipated that the ColSoft programme will be funded to an amount of approximately R3 000 000 per annum, which sum includes the expected support from participating industry partners.

## **Revision History**

**Version 1** prepared by the Western Cape Region under Tony Krzesinski and Edwin Blake, January 1994.

**Version 2a** prepared by Judy Bishop with help from Peter Clayton, February 1994.

**Version 2b** prepared by Edwin Blake with input from Roelf van den Heever, February 1994.

**Version 3** prepared by Tony Krzesinski, with help from Edwin Blake and Chris Brink, March 1994.

## Book Review

**Object Data Management: Object-Oriented and Extended Relational Database Systems**, by R G G Cattell, Addison-Wesley, 1991, 318pp, ISBN 0-201-53092-9.

**Reviewer:** RPD Faria, University of Pretoria (rfaria@doslan.cs.up.ac.za)

Newton's third law states that for every action there is an equal but opposite reaction. Roughly interpreted in the realm of computer science, one may argue that more complicated software systems require more complicated implementations that are harder to maintain. Cattell's book discusses the data management issues involved in implementing and maintaining complex software systems.

This goal driven book on data management makes one appreciate the difficulties involved in creating modern applications. Computer aided software engineering (CASE) and computer aided manufacturing (CAM) are examples of a new trend of complex modern applications. Data management systems that satisfy the stringent requirements of these applications are defined as object data management systems (ODMS). Cattell's book provides a specification for ODMS and implementation issues arising from the specification.

The very first page describes the need for ODMS. A detailed thirty page description of the applications that helped inspire the ODMS technology follows these descriptions of ODMS. The particular applications that are addressed fall under the rubric of software engineering, mechanical engineering, electrical engineering and information technology. Cattell basis the requirements for modern data management systems on the requirements exposed by the specifications of these applications. From these requirements a key observation stating that, traditional database management systems cannot satisfy modern applications' requirements is made.

An introductory text on traditional database systems is found in chapter 3. After discussing key concepts of traditional database systems, the popular relational database technology is detailed. Finally, important implementation issues concerning traditional database systems are raised. This chapter fuelled my reservations about using traditional database systems in modern applications.

Armed with the insight gained in the third chapter, it was easy to understand the fourth chapter, focusing on ODMS concepts. This chapter is indeed the crux of the book. It provides a thorough specification of the ODMS technology.

Chapter four's text defines the concept of a database architecture. These architectures describe the different approaches to implement ODMS. The *database programming language* architecture is one of the more promising and exciting architectures discussed. These architectures provide a tight association between programming languages and data management systems meaning that these systems only have one run-time environment. Researchers may find the small section on *Extended functional and semantic database systems* interesting.

Orthogonal to these architectures, the fundamental

concepts that each of the database architectures should support are described. Objects, object aggregation and object relationships are examples of these concepts. After discussing these fundamental concepts, exciting fresh database system concepts such as object versioning are introduced.

The implementation issues involved in implementing ODMS are fleshed out in chapter five. The book presents alternative implementations for each concept discussed in chapter along with a discussion of expected efficiency of each of the alternatives. Cattell's interest in system performance is evident in these discussions. The chapter ends with an interesting performance comparison of traditional database systems with ODMS using the OO1 benchmark. This benchmark indicates the possibility that ODMS offer large performance benefits over traditional database systems. The book does however caution readers to be careful when interpreting toy benchmarks.

Chapter six describes the requirements of an ODMS from a user's point of view. A number of statements and opinions collectively indicate the recommended minimum functionality of an ODMS. Practicing engineers will relish in the fact that these statements and opinions are based on working database systems.

Future directions for ODMS technology rounds off an informative conclusion that summarizes the book contents.

An informative appendix lists database management system prototypes and products. This list will aid practicing engineers to purchase data management systems to suit their particular needs.

Practicing engineers will enjoy Cattell's book. The book is full of fresh ideas that attempts to capture the essence of the rapidly changing database management technology. Furthermore, a succinct summary concludes each chapter. These summaries allow busy readers to scan through the book. The fact that the book is user and goal oriented left an impression on me. This factor assured me that the author is a realistic, market driven individual.

Researchers may find the writing style informal. Furthermore, the use of ambiguous terminology leads to a large number of unclear statements. Even more tragically, I found some of the statements to be incorrect. The book's treatment of fundamental object oriented concepts is untidy if compared to other books, such as that of Rumbaugh et.al.[1]

In conclusion, we may argue that a book covering a complex topic necessitates a complex text. Cattell's book is an exception to this argument. This is a simple book covering complex material. The writing style of the author is however informal making it difficult to formally reason about the contents of the book. The book suits busy people who want to get up to date on the current changes in the rapidly advancing database management field.

1. J Rumbaugh, M Blaha, W Premerlani, F Eddy and W Lorensen. *Object-Oriented Modeling And Design*. Prentice-Hall, 1991.

# LEXICA

*an integrated environment for machine translation*

**LEXICA's capabilities include :**

- Translator's Workbench
- Powerful word-based Post Editor
- Automatic translator
- Integrated on-line dictionaries
- Support for most Windows-based wordprocessors

**LEXICA**  
*is a language  
independent platform*

*New language pairs  
can be developed  
on request*

**LEXICA provides  
unique support for  
languages spoken on the  
African continent, including :**

- Swahili
- Tswana
- Zulu
- Afrikaans
- Portuguese

**as well as :**

- German and French



**Eenheid vir Programmatuur-Ingenieurswese  
Unit for Software Engineering**

*in association with*  
Department of Computer Science  
University of Pretoria, Pretoria 0002 South Africa

Telephone:  
Fax:  
E-mail:

(012) 420-2504  
43-6454  
lexica@cs.up.ac.za





## Notes for Contributors

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research papers. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as Communications or Viewpoints. While English is the preferred language of the journal, papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted in triplicate to the editor.

### Form of Manuscript

Manuscripts for *review* should be prepared according to the following guidelines.

- Use wide margins and 1½ or double spacing.
- The first page should include:
  - title (as brief as possible);
  - author's initials and surname;
  - author's affiliation and address;
  - an abstract of less than 200 words;
  - an appropriate keyword list;
  - a list of relevant Computing Review Categories.
- Tables and figures should be numbered and titled. Figures should be submitted as original line drawings/printouts, and not photocopies.
- References should be listed at the end of the text in alphabetic order of the (first) author's surname, and should be cited in the text in square brackets [1–3]. References should take the form shown at the end of these notes.

Manuscripts accepted for publication should comply with the above guidelines (except for the spacing requirements), and may be provided in one of the following formats (listed in order of preference):

1. As (a)  $\text{\LaTeX}$  file(s), either on a diskette, or via e-mail/ftp – a  $\text{\LaTeX}$  style file is available from the production editor;
2. As an ASCII file accompanied by a hard-copy showing formatting intentions:
  - Tables and figures should be on separate sheets of paper, clearly numbered on the back and ready for cutting and pasting. Figure titles should appear in the text where the figures are to be placed.
  - Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin, if they are not clear in the text.

Further instructions on how to reduce page charges can be obtained from the production editor.

3. In camera-ready format – a detailed page specification is available from the production editor;
4. In a typed form, suitable for scanning.

### Charges

Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect scanning, typesetting, reproduction and other costs. Currently, the minimum rate is R30-00 per final page for  $\text{\LaTeX}$  or camera-ready contributions and the maximum is R120-00 per page for contributions in typed format (charges include VAT).

These charges may be waived upon request of the author and at the discretion of the editor.

### Proofs

Proofs of accepted papers in categories 2 and 4 above will be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Note that, in the case of camera-ready submissions, it is the author's responsibility to ensure that such submissions are error-free. However, the editor may recommend minor typesetting changes to be made before publication.

### Letters and Communications

Letters to the editor are welcomed. They should be signed, and should be limited to less than about 500 words.

Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

### Book reviews

Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

### Advertisement

Placement of advertisements at R1000-00 per full page per issue and R500-00 per half page per issue will be considered. These charges exclude specialized production costs which will be borne by the advertiser. Enquiries should be directed to the editor.

### References

1. E Ashcroft and Z Manna. 'The translation of 'goto' programs to 'while' programs'. In *Proceedings of IFIP Congress 71*, pp. 250–255, Amsterdam, (1972). North-Holland.
2. C Bohm and G Jacopini. 'Flow diagrams, turing machines and languages with only two formation rules'. *Communications of the ACM*, 9:366–371, (1966).
3. S Ginsburg. *Mathematical theory of context free languages*. McGraw Hill, New York, 1966.

## Contents

### GUEST CONTRIBUTIONS

Ideologies of Information Systems and Technology <b>LD Introna</b> . . . . .	1
What is Information Systems? <b>TD Crossman</b> . . . . .	7

---

### RESEARCH ARTICLES

Intelligent Production Scheduling: A Survey of Current Techniques and An Application in The Footwear Industry <b>V Ram</b> . . . . .	11
Effect of System and Team Size on 4GL Software Development Productivity <b>GR Finnie and GE Wittig</b> . . . . .	18
EDI in South Africa: An Assessment of the Costs and Benefits <b>G Harrington</b> . . . . .	26
Metadata and Security Management in a Persistent Store <b>S Berman</b> . . . . .	39
Markovian Analysis of DQDB MAC Protocol <b>F Bause, P Kritzinger and M Sczittnick</b> . . . . .	47

---

### TECHNICAL NOTE

An evaluation of substring algorithms that determine similarity between surnames <b>GdeV de Kock and C du Plessis</b> . . . . .	58
--	----

---

### COMMUNICATIONS AND REPORTS

Ensuring Successful IT Utilisation in Developing Countries <b>BR Gardner</b> . . . . .	63
Information Technology Training in Organisations: A Replication <b>R Roets</b> . . . . .	68
The Object-Oriented Paradigm: Uncertainties and Insecurities <b>SR Schach</b> . . . . .	77
A Survey of Information Authentication Techniques <b>WB Smuts</b> . . . . .	84
Parallel Execution Strategies for Conventional Logic Programs: A Review <b>PEN Lutu</b> . . . . .	91
The FRD Special Programme on Collaborative Software Research and Development: Draft Call for Proposals . . . . .	99
Book review . . . . .	102

---